CLAIMS

What is claimed is:

1. A method, comprising:

enabling a processor to access first and second memory stores;

employing a first portion of the first memory store as a general-purpose memory store;

employing a second portion of the first memory store as a memory cache for the second memory store.

2. The method of claim 1, wherein the first memory store comprises an off-chip static random access memory (SRAM) memory store.

3. The method of claim 1, wherein the second memory store comprises an off-chip dynamic random access memory (DRAM) memory store.

4. The method of claim 1, wherein the second memory store comprises an off-chip memory store comprising one of Rambus dynamic random access memory (RDRAM), RLDRAM (reduced latency DRAM), DDR (double data rate), DDR-2, or DDR-3 DRAM, and FCDRAM (fast cycle DRAM).

5. The method of claim 1, further comprising:

employing a hardware-based cache management component to manage access to the memory cache.

6. The method of claim 5, further comprising:

maintaining a cache tag array implemented in the hardware-based cache management component.

7. The method of claim 5, wherein the hardware-based cache management component includes a content addressable memory (CAM) component, the method further comprising:

managing access to the memory cache via the CAM component.

8. The method of claim 1, further comprising:

partitioning the memory cache into at least one data cache and at least one address tag array.

9. The method of claim 8, wherein the processor supports concurrent execution of a plurality of threads, the method further comprising:

maintaining an address tag array for each of said plurality of threads; and

maintaining a data cache for each of said plurality of threads.

10. The method of claim 1, wherein the processor comprises a network processor.

11. The method of claim 10, further comprising:

integrating a front-side bus controller on the network processor; and

coupling a memory store to the network processor via a front-side bus.

12. The method of claim 10, further comprising:

employing the second memory store as a memory store for a general-purpose processor integrated into the network processor.

13. A method, comprising:

enabling a plurality of network processors to access first and second shared memory stores;

employing a first portion of the first shared memory store as a memory store;

employing a second portion of the first shared memory store as a memory cache for the second shared memory store.

14. The method of claim 13, wherein the first shared memory store comprises static random access memory (SRAM).

15. The method of claim 13, wherein the second shared memory store comprises dynamic random access memory (DRAM).

16. The method of claim 13, further comprising:

configuring the memory cache as a cache array including a plurality of cache lines; and

maintaining a local address-to-cache line map on each of the plurality of network processors in which cache lines associated with that network processor are mapped to corresponding memory addresses in an address space for the second shared memory store.

17. The method of claim 16, further comprising:

broadcasting a cache line access request to the plurality of network processors in response to a memory access request from a requesting network processors; and

performing a cache line lookup in the local address-to-cache line map of each network processor to determine which, if any, network processors owns a cache line corresponding to the memory access request.

18. The method of claim 17, further comprising:

accessing the cache line to service the memory access request if it is determined that one of the network processors owns the cache line; and

modifying entries in the local address-to-cache line maps for each of the network processor that owned the cache line and the requesting network processor to reassign ownership of the cache line to the requesting network processor.

19. The method of claim 17, further comprising:

accessing memory from the second shared memory store if it is determined that none of the network processors owns the cache line;

selecting a cache line in the memory cache to replace;

copying data corresponding to data contained in the memory in the second shared memory store that is accessed to the cache line selected to be replaced; and

modifying the local address-to-cache line map of the requesting network processor to assign ownership of the cache line that is replaced to the requesting network processor.

20. A network processor, comprising:

an internal interconnect;

a plurality of packet processing micro-engines, each coupled to the internal interconnect;

a static random access memory (SRAM) controller, coupled to the internal interconnect, to control access to an off-chip SRAM memory store;

a general-purpose processor, coupled to the internal interconnect; and

a cache management component, coupled to the internal interconnect, to effectuate a memory cache in a first portion of the off-chip SRAM memory store,

wherein a second portion of the off-chip SRAM memory store may be accessed as a non-cached memory store.

21. The network processor of claim 20, further comprising:

a front-side bus (FSB) interface; and

an FSB controller, coupled between the FSB interface and the internal interconnect,

wherein the cache management component is programmed to manage a memory cache in the SRAM memory store corresponding to a memory address space for a dynamic random access memory (DRAM) memory store to be accessed via the FSB interface.

22. The network processor of claim 20, further comprising:

a private communication channel linking the SRAM controller to the cache management component.

23. The network processor of claim 20, further comprising:

a Rambus dynamic random access memory (RDRAM) interface; and

an RDRAM controller, coupled between the RDRAM interface and the internal interconnect,

wherein the cache management component is programmed to manage a memory cache in the SRAM memory store corresponding to a memory address space for a RDRAM memory store to be accessed via the RDRAM interface.

24. The network processor of claim 20, wherein the cache management component includes facilities for maintaining a hardware-based cache tag array.

25. The network processor of claim 20, wherein the cache management component includes facilities to manage a cache tag array in a portion of the SRAM memory store.

26. The network processor of claim 20, wherein the cache management component includes facilities to effectuate a content addressable memory (CAM) cache management scheme.

27. A system, comprising:

a static random access memory (SRAM) memory store, coupled to an SRAM interface;

a dynamic random access memory (DRAM) memory store, coupled to a front-side bus (FSB); and

a network processor, including:

an internal interconnect;

a plurality of packet processing micro-engines, each coupled to the internal interconnect;

a static random access memory (SRAM) controller, coupled between the internal interconnect and the SRAM interface;

a general-purpose processor, coupled to the internal interconnect;

an FSB interface, coupled to the front-side bus;

an FSB controller, coupled between the internal interconnect and the FSB interface; and

a cache management component, coupled to the internal interconnect, to effectuate a memory cache in a first portion of SRAM memory store, wherein a second portion of the SRAM memory store may be accessed as a non-cached memory store.

28. The system of claim 27, further comprising:

a Rambus dynamic random access memory (RDRAM) memory store, coupled to an RDRAM interface; and

an (RDRAM) controller, integrated on the network processor and coupled between the RDRAM interface and the internal interconnect.

29. The system of claim 27, wherein the FSB controller and the cache management component are integrated.

30. The system of claim 27, wherein the cache management component includes facilities for maintaining a hardware-based cache tag array.